

Evaluation Performance Neural Network Genetic Algorithm

Hetty ROHAYANI¹, Tuga MAURITSIUS², Spit Warnars Harco Leslie H³
and Edi ABDURRACHMAN⁴

¹*Departement of Computer Science, Graduate Program Doctor of Computer Science, Bina Nusantara University, Indonesia, Faculty Techniques and Computer Science, Adiwangsa Jambi University, Indonesia*

²*Departement of Computer Science, Graduate Program Doctor of Computer Science, Bina Nusantara University, Indonesia*

³*Departement of Computer Science, Graduate Program Doctor of Computer Science, Bina Nusantara University, Indonesia*

⁴*Departement of Computer Science, Graduate Program Doctor of Computer Science, Bina Nusantara University, Indonesia*

**Corresponding author: hetty_mna@yahoo.com*

Abstract

Hybrid models of the precipitation spillover process that are embedded with high unpredictability, non-stationarity, and non-linearity in both spatial and worldly scales can give significant results in rainfall forecasting. Considering this, various neural network models have been applied to reproduce this complex process. Neural Network is an information processing system that has characteristics similar to biological terms. A neural network is a machine designed to model the workings of the human brain in performing certain functions or tasks. In general, FFNNs are trained to use the Backpropagation algorithm to get their weights. Backpropagation can work well on simple training problems, but its performance will decrease and be trapped in a local minimum when applied to data that has enormous complexity. Therefore, metaheuristic operations are needed using Genetic Algorithms (AG). In this paper, a detailed discussion of the FFNN-AG step construction will be given, which is a search algorithm based on selection and genetic mechanisms to determine global optimum and evaluation of previous paper related to this.

Keywords: *feed forward neural network, evaluation, rainfall*

Introduction

Weather prediction is essential in planning daily life, one of which is to make a decision. The success of a weather prediction will have an impact on decision making in various fields, including agriculture and aviation [1]. In the field of weather prediction agriculture is used to be able to determine the agricultural schedule to provide the desired results. Estimation of rainfall in the agricultural sector has now become a significant need [2], such as the selection of seeds, fertilizers, and eradication of pests. Information about a lot of rainfall is handy for farmers in anticipating extreme events such as droughts and floods [3].

Whereas in the field of aviation, weather prediction is important to determine the time, location, direction of motion, altitude and plan for aircraft movements to take into account operating disruptions which can be caused if the weather is bad and also to consider determining flight routes or determining when carrying additional in one case the plane had to return due to weather conditions that were not possible [4]

There are two weather prediction methods generally used, namely a method with an empirical approach and the second is a dynamic approach method. The empirical

approach method is often referred to as analog forecasting [5], and this method uses past weather data to predict future events [6]. While the second method is based on equations and is often referred to as a numerical method. Therefore, fast and accurate rainfall estimation is needed [7]. By using a computing system in the field of Artificial Intelligence. The methods used are FFNN, GRNN, BPNN. The identification of patterns of data from the rainfall estimation system can be done using the learning approach method [8]. Based on the learning ability it possesses, then ANN can be trained to study and analyze past data patterns and try to find a formula or function that will link past data patterns with desired output at this time. Previous research determined [9] backpropagation neural network architecture (initial weight and initial bias) using genetic algorithms to solve problems in determining parameters of artificial neural network architecture (initial weight and initial bias) and produce better results than without optimization [10]. The combination of AG and FFNN is a combined form of neural network algorithms and evolutionary algorithms [11]. The algorithm used in the revolutionary algorithm 3 is AG, which combines both algorithms to produce maximum connections and weights. Combined AG and FFNN methods will create connections and weights that are partially related so that they can predict rainfall [12]. John Holland developed genetic

Algorithms in 1970 [13]. Genetic Algorithms are optimization algorithms that mimic the principles[14] of natural genetics and natural selection[15]. Genetic Algorithms search in random directions using the alternatives that have been given to find the best replacement by adjusting to fitness criteria. Fitness is a value that will be maximized or minimized. An early population of chromosomes is taken to produce offspring that can survive to make the next generation. The basic operations of Genetic Algorithms are selection, crossover, and mutation. The remainder of the paper is organized as follows. Section II explains the step construction; Section III presents the evaluation performance. Finally, conclusions and future research directions are indicated in Section IV.

THE STEP CONSTRUCTION

The use of Genetic Algorithms (AG) as a learning method on Neural Network (NN) is an integrated network training procedure to obtain optimum weight or parameters to produce output according to the specified target by minimizing errors [16]. In Feed Forward Neural Network (FFNN) modeling for time series data, the input model is past data ($X_{t-1}, X_{t-2}, \dots, X_{t-p}$) and the target is current data (X_t). The general form of the FFNN model for time series data is written in the following equation (1):

$$X_t = \psi_0 \{ v_{b0} + \sum_{n=1}^H v_{no} \psi_n (w_{bn} + \sum_{i=1}^p w_{in} X_{t-i}) \} \quad (1)$$

With ($w_{bn}, w_{in}, v_{b0}, v_{no}$) is the value of weights or parameters on FFNN while (ψ_n, ψ_0) is the activation function used in each processing unit. Using a specified interval, if the lower limit is r_b and the upper limit is r_a , the code of a gene g and the code bits that make up a gene g_1, g_2, \dots, g_n decoding can be done in the following way.

Real-number encoding

$$x = r_b + (r_a - r_b)g \quad (2)$$

Discrete decimal encoding

$$x = r_b + (r_a - r_b)(g_1 \times 10^{-1} + g_2 \times 10^{-2} + \dots + g_n \times 10^{-n}) \quad (3)$$

Binary encoding

$$x = r_b + (r_a - r_b)(g_1 \times 2^{-1} + g_2 \times 2^{-2} + \dots + g_n \times 2^{-n}) \quad (4)$$

If in the scheme of encoding the discrete decimal encoding and binary encoding the solution sought turns out to be outside the upper limit of the given interval, the decoding formula needs to be changed to

Discrete decimal encoding

$$x = r_b + \left(\frac{r_a - r_b}{\sum_{i=1}^n 10^{-i}} \right) (g_1 \times 10^{-1} + g_2 \times 10^{-2} + \dots + g_n \times 10^{-n}) \quad (5)$$

Binary encoding

$$x = r_b + \left(\frac{r_a - r_b}{\sum_{i=1}^n 2^{-i}} \right) (g_1 \times 2^{-1} + g_2 \times 2^{-2} + \dots + g_n \times 2^{-n}) \quad (6)$$

For a function h which has a small variance, all individuals will have almost the same fitness value. However, it has a negative effect on the selection process which chooses parents proportionally according to the fitness value. For that, we need a mechanism called Linear Fitness Ranking (LFR).

$$f(i) = (N + 1 - R(i)) \quad (7)$$

But the use of fitness values in equation (7) can result in evolution reaching local optima because of the small differences in fitness values for all individuals in the population. The tendency to converge to local optima can be reduced by using equations:

$$f(i) = f_{max} - (f_{max} - f_{min}) \left(\frac{R(i)-1}{N-1} \right) \quad (8)$$

Thus, obtained a fitness value that is in the interval [f_{min}, f_{max}]. The steps for using AG in training FFNN are as follows:

Step 1 Population Initialization

The initialization procedure is to generate several individuals or chromosomes randomly as the initial population by using binary encoding schemes. Based on the FFNN network architecture, the estimated number of weights or parameters is $[H \times (p + 2)] + 1$.

Step 2 Decode of Chromosomes

Each parameter in the chromosome initialized with binary encoding is converted into the real value as in equation (2). The purpose of this chromosome decoding is to make it easier to calculate the fitness value of each chromosome.

Step 3 Individual Evaluation

Determines the fitness value of each chromosome that has been initialized. In this case, the optimization used is minimizing the optimization function so to determine the fitness value of each chromosome determined by the formula $f = 1 / RMSE$.

Step 4 Elitism

Making a copy and storing a chromosome or individual has the best fitness value from a population that has been initialized.

Step 5 Linear Fitness Ranking (LFR)

Scaling fitness values obtained from individual evaluations. LFR is used to avoid the tendency to converge on local optimal solutions by getting a new fitness value that has a greater variance (Equation 8).

Step 6 Selection

Choose two chromosomes or individuals to be used as parent pairs. The selection method used is the tournament selection.

Step 7 Crossover

Transfer a portion of the chromosome to the chromosome of the pair, and vice versa so that two new chromosomes are produced. This process occurs if a random number appears less than the specified crossing probability.

Step 8 Mutation

Changing one or more genetic codes on each chromosome becomes inverse. This process occurs if a random number that appears is less than the mutation probability (p_m) specified.

Step 9 Population Replacement (Generational Replacement)

Replacing members of the initial population as a result of initialization with a new population consisting of the chromosomes produced by elitism, crossbreeding and mutations. Re-do steps 2 to 4 so that the maximum fitness value is obtained, if it has not been obtained, steps 5-9 are done. This new population will be the second generation population to be processed in step 10.

Genetic Algorithm itself is used to optimize the weight and bias of a randomized neural network at the beginning. GA will find the best network model generated from critical GA processes. GA will do generate weight, and initial bias using important processes from GA and uses

an error value from Back Propagation Neural Network to measure network model performance produced. Network with the smallest error value is the best network model that will be used for process testing. In the backpropagation phase, each the output unit accepts the target pattern associated with the pattern input for calculated error value. This error will be propagated backward. While the weight modification phase aims to reduce errors that occur. The third phase it is repeated continuously until the termination condition fulfilled. The popular network architecture used in the FFNN model is a multilayer network consisting of input layers, hidden layers, and output layers. In determining the number of units in the input layer, there is no standard provision. So is the number of units in the hidden layer and the output layer. Some studies are limited to the number of hidden layer units equal to the number of units in the input. After the network architecture is formed on FFNN, the next step is to find estimates of the weights or parameters on FFNN using Genetic Algorithms (AG). The entire procedure for using AG in FFNN training can be presented merely with the figure 1 flow diagram as follows.

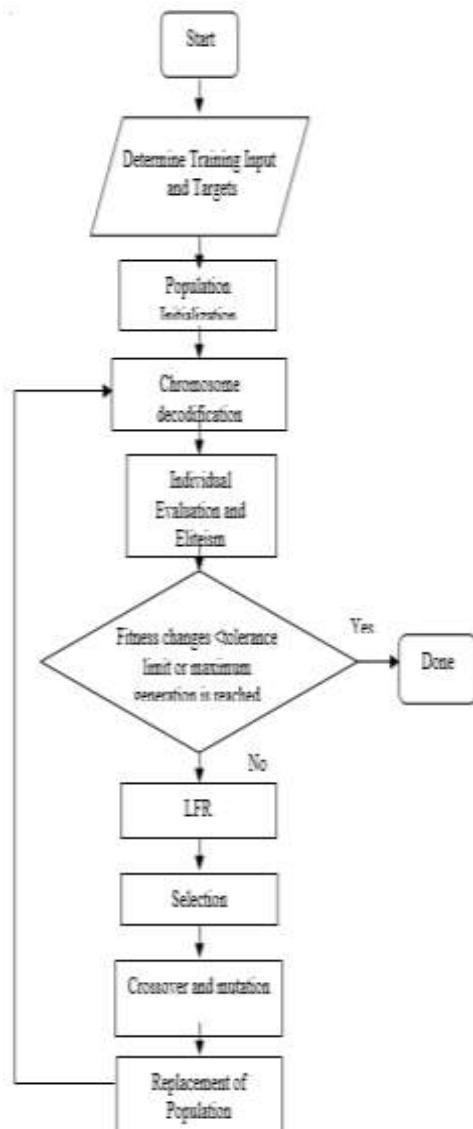


Figure 1. Step Construction of FFNN AG

The procedure of AG as a training method in FFNN is done by initializing the population as a solution space

containing chromosomes. Each chromosome represents several genes which when decoded will produce network weights or parameters. Furthermore, the chromosomes will be evaluated by using a specific objective function to get the fitness value which is the measurement scale in AG. To get a new generation, screening was done based on fitness values with a certain selection method and using crossover and mutation genetic operators. After several generations have been determined, the AG will converge to the best chromosome, and the solution is obtained in the form of the optimum weight value or parameter. For example, if a population size of 10 chromosomes is determined, each gene in the chromosome contains 10 bits, the probability of crossing (p_c) = 0.8, probability of mutation (p_m) = 0.01, tournament selection method with size $k= 4$ and tournament probability (p_k) = 0.8. Multilayer networks are an extension of a single layer network. Multilayer networks consist of input layers, hidden layers, and output layers. The hidden layer is located between the input layer and the output layer. The output of a hidden layer will be input for the next layer. This network has at least one hidden layer. The architecture of a multilayer network is described as follows:

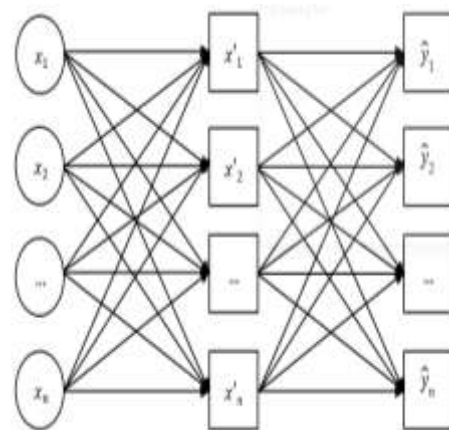


Figure 2. Multilayer Neural Network

Evaluation Performance

The use of FFNN metaheuristic has been published on science direct platforms in 1995 to 2018 as many as 7343 articles with the distribution of categories in Figure 3

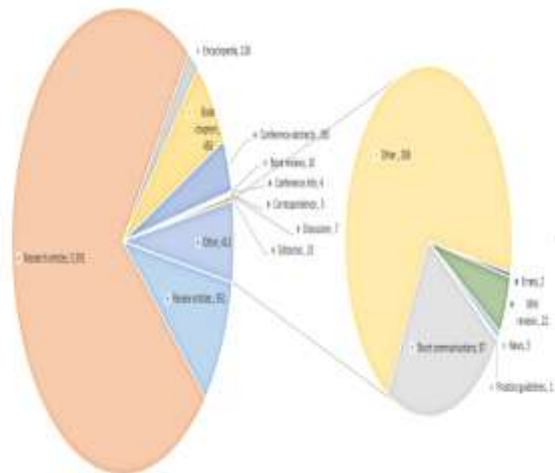


Figure 3. Systematic Review

Commonly, model performance was evaluated by the mean square error (MSE), the root mean square error

(RMSE), the normalized root mean square error (NRMSE), and the mean absolute percentage error (MAPE). Moreover, The MSE quantifies the differences between observed and predicted values along the time series axis and penalizes for the large differences because

of its square power. This allows the ANN models to better capture peak values of time series data. The MSE is commonly used as the training objective to be minimized in ANN models

Table 1. The Description of selected studies

Author	Results	Performance
Asadi et al (2013) [17]	Levenberg–Marquardt neural networks (PELMNN) model for runoff prediction by combining genetic algorithms and feed forward neural networks	Calibration RMSE= 0.029 Verification = 0.036
Nasseri et al. (2008) [18]	An ANN procedure (MFNN) integrated with an evolutionary optimization method such as GA was applied	NMSE= 0.002
Wu et al (2015) [19]	Effective hybrid optimization strategy by incorporating the adaptive optimization of particle swarm optimization (PSO) into genetic algorithm (GA), namely HPSOGA, is used for determining the parameters of radial basis function neural networks (number of neurons, their respective centers and radii) automatically	RMSE RBF-GA 111.9055 RMSE RBF-HPSOGA 67.7286
Mislan et al. (2015) [20]	A BPNN algorithm has been used to model and predict rainfall in Tenggarong, East Kalimantan - Indonesia. After testing the three architectures with different epochs; 500, 1000 and 1500,	MSE = 0.00096341
Kasiviswanathan et al. (2013) [21]	artificial neural network (ANN) rainfall runoff models during calibration with a consideration of generating ensemble predictions. In Stage 1, ANN model is trained with genetic algorithm (GA) to obtain optimal set of weights and biases vector. In Stage 2, possible variability of ANN parameters (obtained in Stage 1) is optimized so as to create an ensemble of models with the consideration of minimum residual variance for the ensemble mean.	Stage 1 ensemble RMSE = 31.55 Stage 1 Validation RMSE = 89.16
Danandeh Mehr, and Nourani(2017) [22]	a new hybrid model that integrates moving average filtering with multigene GP and uses Pareto-front plot to optimize the evolved models through an interactive complexity-efficiency trade-off	the RMSE and PI measures at testing period were improved approximately 70% and 76%
Nourani, et al (2013) [23]	two-level self-organizing map (SOM) was utilized to distinguish spatially homogeneous bunches of precipitation satellite information, and to pick the most usable and powerful information for a feed-forward neural organize (FFNN) to show precipitation overflow process on a day by day and multi-step ahead time scale. The wavelet change (WT) was likewise used to separate dynamic and multi-scale highlights of the non-stationary spillover time arrangement and to expel clamor. The exhibition of the coupled SOM–FFNN model was contrasted with the recently proposed joined SOM–WT–FFNN model.	Validation of the SOM–FFNN and SOM–WT–FFNN models were 0.80 and 0.93, respectively

However, the limitations of MSE and E are that they are unable to provide the practical assessment of difference between predicted and observed values. For example, MSE provides similar value of 4 for the difference between observation of 1000 and prediction of 1002 as well as between observation of 1 and prediction of 3 in which it is obviously that the former prediction is very

good while the latter prediction is very poor. We can use data sets from covariate and target $\{X_s, t_s\}$, where $s = 1, 2, \dots, N$. The activation function of the output can be used linear function and can minimize E errors

$$E = \frac{1}{2} \sum_{s=1}^N (y(x_s, \omega) - t_s)^2 \quad (9)$$

Where $y(x_s, \omega)$ is the output of each observation and can also be said to be part of the probability. If the target of us follows a Gaussian distribution, the output will be

$$p(t|x, \omega) \approx \mathcal{N}\left(t|y(x, \omega) - \frac{1}{\beta}\right)^2 \quad (10)$$

Where β is inverse from error which follow a Gaussian distribution. In the above equation, we can get the likelihood function for weight estimation ω

$$p(t|x, \omega) = \prod_{s=1}^N p(t_s|x_s, \omega, \beta) \quad (11)$$

To evaluate the model then

$$a_j = \sum_i w_{ji} z_i \quad (12)$$

$$z_i = h(a_j) \quad (13)$$

Where w_{ji} is any weight containing information that has passed every step of the neuron, from i to j neurons and z_i is the output of neurons and $h()$ is an activation function.

To update the weighting, you must meet the rule chain and fulfill $\frac{\partial a_j}{\partial \omega_{ji}} = z_j$ as follows:

$$\frac{\partial E}{\partial \omega_{ji}} = \frac{\partial E}{\partial a_j} \frac{\partial a_j}{\partial \omega_{ji}} \quad (14)$$

$$\frac{\partial E}{\partial \omega_{ji}} = \delta_j z_j \quad (15)$$

$$\delta_j = h'(a_t) \frac{\partial E}{\partial y_t} \quad (16)$$

For neurons in the output layer. Where t is the number of neurons and y_t is the output of neurons in the output layer. So that the neurons in the hidden layer δ_j can be written as follows

$$\delta_j = h'(a_t) \sum_t \omega_{tj} \delta_t \quad (17)$$

$$\frac{\partial E}{\partial \omega_{ji}} = \sum_{s=1}^N \frac{\partial E_s}{\partial \omega_{ji}} \quad (18)$$

Our main focus is to find the weight vector ω and minimize error $E(\omega)$. So we need a gradient of error and weighting for each point ($\nabla E = 0$)

$$\text{gradient} = \nabla E = \mathbf{H}(\mathbf{w} - \mathbf{w}^*) \quad (19)$$

Where \mathbf{H} is the hessian matrix obtained from the second derivative of the weighting error function $\mathbf{H} = \frac{\partial^2 E}{\partial \omega_i \partial \omega_j}$

$$\mathbf{w}^* = \mathbf{w} - \mathbf{H}^{-1} \nabla E \quad (20)$$

$\mathbf{H}^{-1} \nabla E$ are newton's direction and hessian matrix can be generated from $G^{(k)}$

$$G^{(k+1)} = G^{(k)} + \frac{pp^T}{p^T v} - \frac{(G^{(k)} v) v^T G^{(k)}}{v^T G^{(k)} v} + (v^T G^{(k)} v) u u^T \quad (21)$$

$$p = w^{k+1} - w^k \quad (22)$$

$$v = g^{k+1} - g^k \quad (23)$$

$$u = \frac{p}{p^T v} - \frac{G^{(k)} v}{v^T G^{(k)} v} \quad (24)$$

$$w^{k+1} = w^k + \zeta^k G^{(k)} g^T \quad (25)$$

Weather prediction means predicting how the current state of the atmosphere will change. If you want to make weather predictions, the current weather conditions in an extensive area must be known. The information is obtained, a network of weather observation stations is placed throughout the world. More than 10,000 weather stations and hundreds of ships provide surface weather information four times a day. Most airports observe weather conditions every hour, especially weather data above the upper air obtained from radiosonde, airplanes, and satellites. In the mid-1950s, all maps and weather charts were plotted by hand and analyzed individually. Meteorologists make predictions using specific rules related to the particular weather system in question. Short-term weather prediction is less than 6 hours, and the surface weather system will move at a stable speed. The upper air graph will be used to predict where the storm surface will form and where the upper air pressure system will increase or weaken.

Predicted positions on this system will be extrapolated to the future using linear graph techniques and weather maps. Experience will play an essential role in making predictions. In some cases, predictions with this method will be accurate, and in general, it is quite good. But, along with the advancement of modern computers, as well as increasingly advanced observation techniques, the current prediction method is better. One of the weather prediction methods used is numerical weather prediction. Weather prediction is a new scientific discipline that developed fundamentally in the second half of the twentieth century by providing many benefits continuously in the field of computing. Modern computers can analyze data that amounts to very much with a speedy time. Every day, data obtained from thousands of observations will be transmitted and processed using high-speed computers. Weather prediction using a computer is called numerical weather prediction. The technique that has been applied is used to solve the equation of atmospheric behavior, which is used to compute numerical values of the climatic characteristics of the future from the initial value, called meteorological observation.

CONCLUSION

The application of genetic algorithms in the optimization of artificial neural networks is made to get the structure of neurons in the hidden layer that is near optimal. The level of recognition of NN in high estimation will be obtained if all neurons in the hidden layer provide a high objective contribution value. The workings of FFNN training using Genetic Algorithms for prediction is to form a network architecture on FFNN, then after the network architecture is developed the next step is to find weight estimates or parameters using Genetic Algorithms (AG). The procedure of AG as a training method on FFNN is done by initializing the population, then decoding chromosomes, evaluating individuals and elitism. If the maximum generation has not been achieved will be Linear Fitness Ranking (LFR), selection, crossover, mutation, population replacement, then individual evaluation and elitism will be carried out until the specified maximum generation is reached. Genetic Algorithms can be used to optimize the work of ANN. the structure of neurons in the hidden layer to obtain the best model measured by the RMSE value obtained. The division of data groups into training data, and testing data gives relatively different results.

REFERENCES

- [1] R. E. Caraka and M. Tahmid, *Statistika Klimatologi*, 1st ed. Yogyakarta: Mobius Graha Ilmu, 2019.
- [2] G. Airlangga, A. Rachmat, and D. Lapihu, "Comparison of exponential smoothing and neural network method to forecast rice production in Indonesia," *Telkomnika*, vol. 17, no. 3, pp. 1367–1375, 2019.
- [3] BMKG, "Essentials of Meteorology," *Goodreads*. 2015.

- [4] R. E. Caraka, S. A. Bakar, M. Tahmid, H. Yasin, and I. D. Kurniawan, "Neurocomputing Fundamental Climate Analysis," *Telkonnika*, vol. 17, no. 4, pp. 1818–1827, 2019.
- [5] R. E. Caraka *et al.*, "The step construction of penalized spline in electrical power load data," *Telkonnika*, vol. 17, no. 2, 2019.
- [6] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting & Control*. 1994.
- [7] D. D. Prastyo, F. S. Nabila, Suhartono, M. H. Lee, N. Suhermi, and S. F. Fam, "VAR and GSTAR-based feature selection in support vector regression for multivariate spatio-temporal forecasting," in *Communications in Computer and Information Science*, 2019, pp. 46–57.
- [8] R. E. Caraka, G. Darmawan, B. D. Supatmato, and B. Hutapea, "Generalized Spatio Temporal Autoregressive Rainfall-Enso Pattern In East Java Indonesia," in *IEEE*, 2018, pp. 75–79.
- [9] J. N. D. Gupta and R. S. Sexton, "Comparing backpropagation with a genetic algorithm for neural network training," *Omega*, 1999.
- [10] H. Iba and C. C. Aranha, "Introduction to genetic algorithms," *Adaptation, Learning, and Optimization*, 2012.
- [11] T. Toharudin, R. E. Caraka, A. Bachrudin, S. A. Bakar, and D. Ambarsari, "Copula-Based Feedforward Neural Network Genetic Algorithm Cargo Forecasting," *Int. J. Adv. Soft Comput. Its Appl.*, vol. 11, no. 1, pp. 22–23, 2019.
- [12] K. W. Wong, P. M. Wong, T. D. Gedeon, and C. C. Fung, "Rainfall prediction model using soft computing technique," *Soft Comput.*, 2003.
- [13] M. Kubat, "The Genetic Algorithm," in *An Introduction to Machine Learning*, 2017.
- [14] O. Kramer, *Genetic Algorithm Essentials*. 2017.
- [15] C. Algorithm, "What Is the Genetic Algorithm?," *Diversity*, 2008.
- [16] Y. Hasbi, W. Budi, and S. Rukun, "Feed Forward Neural Network Modeling for Rainfall Prediction," *E3S Web Conf.*, 2018.
- [17] S. Asadi, J. Shahrabi, P. Abbaszadeh, and S. Tabanmehr, "A new hybrid artificial neural networks for rainfall-runoff process modeling," *Neurocomputing*, 2013.
- [18] M. Nasser, K. Asghari, and M. J. Abedini, "Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network," *Expert Syst. Appl.*, 2008.
- [19] J. Wu, J. Long, and M. Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," *Neurocomputing*, 2015.
- [20] Mislan, Haviluddin, S. Hardwinarto, Sumaryono, and M. Aipassa, "Rainfall Monthly Prediction Based on Artificial Neural Network: A Case Study in Tenggarong Station, East Kalimantan - Indonesia," in *Procedia Computer Science*, 2015.
- [21] K. S. Kasiviswanathan, R. Cibin, K. P. Sudheer, and I. Chaubey, "Constructing prediction interval for artificial neural network rainfall runoff models based on ensemble simulations," *J. Hydrol.*, 2013.
- [22] A. Danandeh Mehr and V. Nourani, "A Pareto-optimal moving average-multigene genetic programming model for rainfall-runoff modelling," *Environ. Model. Softw.*, 2017.
- [23] V. Nourani, A. H. Baghanam, J. Adamowski, and M. Gebremichael, "Using self-organizing maps and wavelet transforms for space-time pre-processing of satellite precipitation and runoff data in neural network based rainfall-runoff modeling," *J. Hydrol.*, vol. 476, pp. 228–243, 2013.